fc	fractional capability variable
$x,\ g,\ a,\ b$	expression variable
k	integer variable
el	array-element variable
l	location variable
M	matrix variable

m	$::=$			matrix expressions
	$\|$	M_{k_1,k_2}		matrix variables
	$\|$	$m + m'$		matrix addition
	$\|$	$m\, m'$		matrix multiplication
f	$::=$			fractional capability
	$\|$	fc		variable
	$\|$	1		whole capability
	$\|$	$\frac{1}{2} \cdot f$		
t	$::=$			linear type
	$\|$	**unit**		unit
	$\|$	**bool**		boolean (true/false)
	$\|$	**int**		63-bit integers
	$\|$	**elt**		array element
	$\|$	f **arr**		arrays
	$\|$	f **mat**		matrices
	$\|$	$!t$		multiple-use type
	$\|$	$\forall\, fc.t$	bind fc in t	frac. cap. generalisation
	$\|$	$t \otimes t'$		pair
	$\|$	$t \multimap t'$		linear function
	$\|$	(t)	S	parentheses
p	$::=$			primitive
	$\|$	**not**		boolean negation
	$\|$	$(+)$		integer addition
	$\|$	$(-)$		integer subtraction
	$\|$	$(*)$		integer multiplication
	$\|$	$(/)$		integer division
	$\|$	$(=)$		integer equality
	$\|$	$(\langle)$		integer less-than
	$\|$	$(+.)$		element addition
	$\|$	$(-.)$		element subtraction
	$\|$	$(*.)$		element multiplication
	$\|$	$(/.)$		element division
	$\|$	$(= .)$		element equality
	$\|$	$(< .)$		element less-than
	$\|$	**set**		array index assignment
	$\|$	**get**		array indexing
	$\|$	**share**		share array
	$\|$	**unshare**		unshare array
	$\|$	**free**		free arrary
	$\|$	**array**		Owl: make array
	$\|$	**copy**		Owl: copy array
	$\|$	**sin**		Owl: map sine over array
	$\|$	**hypot**		Owl: $x_i := \sqrt{x_i^2 + y_i^2}$
	$\|$	**asum**		BLAS: $\sum_i \|x_i\|$

	|	**axpy**		BLAS: $x := \alpha x + y$
	|	**dot**		BLAS: $x \cdot y$
	|	**rotmg**		BLAS: see its docs
	|	**scal**		BLAS: $x := \alpha x$
	|	**amax**		BLAS: $\operatorname{argmax} i : x_i$
	|	**setM**		matrix index assignment
	|	**getM**		matrix indexing
	|	**shareM**		share matrix
	|	**unshareM**		unshare matrix
	|	**freeM**		free matrix
	|	**matrix**		Owl: make matrix
	|	**copyM**		Owl: copy matrix
	|	**copyM_to**		Owl: copy matrix onto another
	|	**sizeM**		dimension of matrix
	|	**trnsp**		transpose matrix
	|	**gemm**		BLAS: $C := \alpha A^{T?} B^{T?} + \beta C$
	|	**symm**		BLAS: $C := \alpha AB + \beta C$
	|	**posv**		BLAS: Cholesky decomp. and solve
	|	**potrs**		BLAS: solve with given Cholesky
v	$::=$			values
	|	p		primitives
	|	x		variable
	|	$()$		unit introduction
	|	**true**		true
	|	**false**		false
	|	k		integer
	|	l		heap location
	|	el		array element
	|	$\mathbf{Many}\, v$		!-introduction
	|	$\mathbf{fun}\, fc \to v$		frac. cap. abstraction
	|	$v[f]$		frac. cap. specialisation
	|	(v, v')		pair introduction
	|	$\mathbf{fun}\, x : t \to e$	bind x in e	abstraction
	|	$\mathbf{fix}\,(g, x : t, e : t')$	bind $g \cup x$ in e	fixpoint
	|	(v)	S	parentheses
e	$::=$			expression
	|	p		primitives
	|	x		variable
	|	$\mathbf{let}\, x = e \,\mathbf{in}\, e'$	bind x in e'	let binding
	|	$()$		unit introduction
	|	$\mathbf{let}\, () = e \,\mathbf{in}\, e'$		unit elimination
	|	**true**		true
	|	**false**		false
	|	$\mathbf{if}\, e \,\mathbf{then}\, e_1 \,\mathbf{else}\, e_2$		if

	$\vert$	k		integer
	$\vert$	l		heap location
	$\vert$	el		array element
	$\vert$	$\textbf{Many}\ e$		!-introduction
	$\vert$	$\textbf{let Many}\ x = e\ \textbf{in}\ e'$		!-elimination
	$\vert$	$\textbf{fun}\ fc \to e$		frac. cap. abstraction
	$\vert$	$e[f]$		frac. cap. specialisation
	$\vert$	(e, e')		pair introduction
	$\vert$	$\textbf{let}\ (a, b) = e\ \textbf{in}\ e'$	$\mathsf{bind}\ a \cup b\ \mathsf{in}\ e'$	pair elimination
	$\vert$	$\textbf{fun}\ x : t \to e$	$\mathsf{bind}\ x\ \mathsf{in}\ e$	abstraction
	$\vert$	$e\ e'$		application
	$\vert$	$\textbf{fix}\ (g, x : t, e : t')$	$\mathsf{bind}\ g \cup x\ \mathsf{in}\ e$	fixpoint
	$\vert$	(e)	S	parentheses
C	$::=$			evaluation contexts
	$\vert$	$\textbf{let}\ x = [-]\ \textbf{in}\ e$	$\mathsf{bind}\ x\ \mathsf{in}\ e$	let binding
	$\vert$	$\textbf{let}\ () = [-]\ \textbf{in}\ e$		unit elimination
	$\vert$	$\textbf{if}\ [-]\ \textbf{then}\ e_1\ \textbf{else}\ e_2$		if
	$\vert$	$\textbf{Many}\ [-]$		!-introduction
	$\vert$	$\textbf{let Many}\ x = [-]\ \textbf{in}\ e$		!-elimination
	$\vert$	$\textbf{fun}\ fc \to [-]$		frac. cap. abstraction
	$\vert$	$[-][f]$		frac. cap. specialisation
	$\vert$	$([-], e)$		pair introduction
	$\vert$	$(v, [-])$		pair introduction
	$\vert$	$\textbf{let}\ (a, b) = [-]\ \textbf{in}\ e$	$\mathsf{bind}\ a \cup b\ \mathsf{in}\ e$	pair elimination
	$\vert$	$[-]e$		application
	$\vert$	$v[-]$		application
Θ	$::=$			fractional capability environment
	$\vert$	$\cdot$		
	$\vert$	Θ, fc		
Γ	$::=$			linear types environment
	$\vert$	$\cdot$		
	$\vert$	$\Gamma, x : t$		
	$\vert$	Γ, Γ'		
Δ	$::=$			intuitionistic types environment
	$\vert$	$\cdot$		
	$\vert$	$\Delta, x : t$		
σ	$::=$			heap
	$\vert$	$\{\}$		empty heap
	$\vert$	$\sigma \uplus \{l \mapsto_f m\}$		location l points to matrix m

$\boxed{\Theta \vdash f\ \mathsf{Cap}}$ Valid fractional capabilities

$$\frac{fc \in \Theta}{\Theta \vdash fc\ \mathsf{Cap}}\ \textsc{WF\_Cap\_Var}$$

$$\frac{}{\Theta \vdash 1\ \mathsf{Cap}}\ \textsc{WF\_Cap\_Zero}$$

$$\frac{\Theta \vdash f \ \mathsf{Cap}}{\Theta \vdash \frac{1}{2} \cdot f \ \mathsf{Cap}} \quad \text{WF\_CAP\_SUCC}$$

$\boxed{\Theta \vdash t \ \mathsf{Type}}$ Valid types

$$\frac{}{\Theta \vdash \mathbf{unit} \ \mathsf{Type}} \quad \text{WF\_TYPE\_UNIT}$$

$$\frac{}{\Theta \vdash \mathbf{bool} \ \mathsf{Type}} \quad \text{WF\_TYPE\_BOOL}$$

$$\frac{}{\Theta \vdash \mathbf{int} \ \mathsf{Type}} \quad \text{WF\_TYPE\_INT}$$

$$\frac{}{\Theta \vdash \mathbf{elt} \ \mathsf{Type}} \quad \text{WF\_TYPE\_ELT}$$

$$\frac{\Theta \vdash f \ \mathsf{Cap}}{\Theta \vdash f \ \mathbf{arr} \ \mathsf{Type}} \quad \text{WF\_TYPE\_ARRAY}$$

$$\frac{\Theta \vdash t \ \mathsf{Type}}{\Theta \vdash !t \ \mathsf{Type}} \quad \text{WF\_TYPE\_BANG}$$

$$\frac{\Theta, fc \vdash t \ \mathsf{Type}}{\Theta \vdash \forall fc.t \ \mathsf{Type}} \quad \text{WF\_TYPE\_GEN}$$

$$\frac{\begin{array}{c}\Theta \vdash t \ \mathsf{Type} \\ \Theta \vdash t' \ \mathsf{Type}\end{array}}{\Theta \vdash t \otimes t' \ \mathsf{Type}} \quad \text{WF\_TYPE\_PAIR}$$

$$\frac{\begin{array}{c}\Theta \vdash t \ \mathsf{Type} \\ \Theta \vdash t' \ \mathsf{Type}\end{array}}{\Theta \vdash t \multimap t' \ \mathsf{Type}} \quad \text{WF\_TYPE\_LOLLY}$$

$\boxed{\Theta; \Delta; \Gamma \vdash e : t}$ Typing rules for expressions

$$\frac{}{\Theta; \Delta; \cdot, x : t \vdash x : t} \quad \text{TY\_VAR\_LIN}$$

$$\frac{x : t \in \Delta}{\Theta; \Delta; \cdot \vdash x : t} \quad \text{TY\_VAR}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : t \\ \Theta; \Delta; \Gamma', x : t \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let} \ x = e \ \mathbf{in} \ e' : t'} \quad \text{TY\_LET}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash () : \mathbf{unit}} \quad \text{TY\_UNIT\_INTRO}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \cdot \vdash e : \mathbf{unit} \\ \Theta; \Delta; \Gamma \vdash e' : t\end{array}}{\Theta; \Delta; \Gamma \vdash \mathbf{let} \ () = e \ \mathbf{in} \ e' : t} \quad \text{TY\_UNIT\_ELIM}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{true} : !\mathbf{bool}} \quad \text{TY\_BOOL\_TRUE}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash \mathbf{false} : !\mathbf{bool}} \quad \text{TY\_BOOL\_FALSE}$$

$$\frac{\begin{array}{c}\Theta; \Delta; \Gamma \vdash e : \mathbf{bool} \\ \Theta; \Delta; \Gamma' \vdash e_1 : t' \\ \Theta; \Delta; \Gamma' \vdash e_2 : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{if} \ e \ \mathbf{then} \ e_1 \ \mathbf{else} \ e_2 : t} \quad \text{TY\_BOOL\_ELIM}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash k : !\mathbf{int}} \quad \text{TY\_INT\_INTRO}$$

$$\frac{}{\Theta; \Delta; \cdot \vdash el : !\mathbf{elt}} \quad \text{TY\_ELT\_INTRO}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \cdot \vdash v : t \\ v \neq l\end{array}}{\Theta; \Delta; \cdot \vdash \mathbf{Many}\ v : !t} \quad \text{TY\_BANG\_INTRO}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : !t \\ \Theta; \Delta, x : t; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let\ Many}\ x = e\ \mathbf{in}\ e' : t'} \quad \text{TY\_BANG\_ELIM}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : t \\ \Theta; \Delta; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash (e, e') : t \otimes t'} \quad \text{TY\_PAIR\_INTRO}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e_{12} : t_1 \otimes t_2 \\ \Theta; \Delta; \Gamma', a : t_1, b : t_2 \vdash e : t\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash \mathbf{let}\ (a, b) = e_{12}\ \mathbf{in}\ e : t} \quad \text{TY\_PAIR\_ELIM}$$

$$\frac{\begin{array}{l}\Theta \vdash t'\ \mathsf{Type} \\ \Theta; \Delta; \Gamma, x : t' \vdash e : t\end{array}}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\ x : t' \to e : t' \multimap t} \quad \text{TY\_LAMBDA}$$

$$\frac{\begin{array}{l}\Theta; \Delta; \Gamma \vdash e : t' \multimap t \\ \Theta; \Delta; \Gamma' \vdash e' : t'\end{array}}{\Theta; \Delta; \Gamma, \Gamma' \vdash e\ e' : t} \quad \text{TY\_APP}$$

$$\frac{\Theta, fc; \Delta; \Gamma \vdash e : t}{\Theta; \Delta; \Gamma \vdash \mathbf{fun}\ fc \to e : \forall fc.t} \quad \text{TY\_GEN}$$

$$\frac{\begin{array}{l}\Theta \vdash f\ \mathsf{Cap} \\ \Theta; \Delta; \Gamma \vdash e : \forall fc.t\end{array}}{\Theta; \Delta; \Gamma \vdash e[f] : t[f/fc]} \quad \text{TY\_SPC}$$

$$\frac{\Theta; \Delta, g : t \multimap t'; \cdot, x : t \vdash e : t'}{\Theta; \Delta; \cdot \vdash \mathbf{fix}\ (g, x : t, e : t') : !(t \multimap t')} \quad \text{TY\_FIX}$$

$\boxed{\langle \sigma, e \rangle \to \langle \sigma', e' \rangle}$ operational semantics

$$\frac{}{\langle \sigma, \mathbf{let}\ () = ()\ \mathbf{in}\ e \rangle \to \langle \sigma, e \rangle} \quad \text{OP\_LET\_UNIT}$$

$$\frac{}{\langle \sigma, \mathbf{let}\ x = v\ \mathbf{in}\ e \rangle \to \langle \sigma, e[x/v] \rangle} \quad \text{OP\_LET\_VAR}$$

$$\frac{}{\langle \sigma, \mathbf{if\ true\ then}\ e_1\ \mathbf{else}\ e_2 \rangle \to \langle \sigma, e_1 \rangle} \quad \text{OP\_IF\_TRUE}$$

$$\frac{}{\langle \sigma, \mathbf{if\ false\ then}\ e_1\ \mathbf{else}\ e_2 \rangle \to \langle \sigma, e_2 \rangle} \quad \text{OP\_IF\_FALSE}$$

$$\frac{}{\langle \sigma, \mathbf{let\ Many}\ x = \mathbf{Many}\ v\ \mathbf{in}\ e \rangle \to \langle \sigma, e[x/v] \rangle} \quad \text{OP\_LET\_MANY}$$

$$\frac{e_1 = e[g/\mathbf{let\ Many}\ g = \mathbf{fix}\ (g, x : t, e : t')\ \mathbf{in\ fun}\ x : t \to e]}{\langle \sigma, \mathbf{let\ Many}\ g = \mathbf{fix}\ (g, x : t, e : t')\ \mathbf{in}\ e' \rangle \to \langle \sigma, e'[g/\mathbf{fun}\ x : t \to e_1] \rangle} \quad \text{OP\_LET\_FIX}$$

$$\frac{}{\langle \sigma, (\mathbf{fun}\ fc \to v)[f] \rangle \to \langle \sigma, v[fc/f] \rangle} \quad \text{OP\_FRAC\_CAP}$$

$$\frac{}{\langle \sigma, (\mathbf{fun}\, x : t \to e)\; v\rangle \to \langle \sigma, e[x/v]\rangle}\quad \textsc{Op\_App}$$

$$\frac{e \to e'}{\langle \sigma, C[e]\rangle \to \langle \sigma, C[e']\rangle}\quad \textsc{Op\_Context}$$

$$\frac{0 \leq k_1, k_2}{\langle \sigma, \mathbf{matrix}\, k_1\; k_2\rangle \to \langle \sigma \uplus \{l \mapsto_1 M_{k_1,k_2}\}, l\rangle}\quad \textsc{Op\_Matrix}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_1 M_{k_1,k_2}\}, \mathbf{free}\, l\rangle \to \langle \sigma, ()\rangle}\quad \textsc{Op\_Free}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_f M_{k_1,k_2}\}, \mathbf{share}\, l\rangle \to \langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\}, (l, l)\rangle}\quad \textsc{Op\_Share}$$

$$\frac{}{\langle \sigma \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\} \uplus \{l \mapsto_{\frac{1}{2}\cdot f} M_{k_1,k_2}\}, \mathbf{unshare}\, l\; l\rangle \to \langle \sigma \uplus \{l \mapsto_f M_{k_1,k_2}\}, l\rangle}\quad \textsc{Op\_Unshare}$$

$$\frac{\begin{array}{c}\sigma(l_1) = M_{k_1,k_2} \\ \sigma(l_2) = M_{k_2,k_3}\end{array}}{\langle \sigma \uplus \{l_3 \mapsto_1 M_{k_1,k_3}\}, \mathbf{gemm}\, l_1\; l_2\; l_3\rangle \to \langle \sigma \uplus \{l_3 \mapsto_1 M_{k_1,k_2}\, M_{k_2,k_3} + M_{k_1,k_3}\}, ((l_1, l_2), l_3)\rangle}\quad \textsc{Op\_Gemm}$$

$$\mathcal{V}[\![\mathbf{unit}]\!] = \{(\{\}, *)\}$$

$$\mathcal{V}[\![!\mathbf{bool}]\!] = \{(\{\}, true), (\{\}, false)\}$$

$$\mathcal{V}[\![!\mathbf{int}]\!] = \{(\{\}, n) \mid 2^{-63} \leq n \leq 2^{63} - 1\}$$

$$\mathcal{V}[\![!\mathbf{elt}]\!] = \{(\{\}, f) \mid f \text{aIEEEFloat64}\}$$

$$\mathcal{V}[\![\mathbf{Z}\,\mathbf{mat}]\!] = \{(\{l \mapsto_{m_W} \_\}, l)\}$$

$$\mathcal{V}[\![(\mathbf{S}f)\mathbf{mat}]\!] = \{(\sigma, l) \mid l \in \mathrm{dom}(\sigma) \land \sigma \star \sigma \in \pi_1[\mathcal{V}[\![f\,\mathbf{mat}]\!]]\}$$

$$\begin{aligned}\mathcal{V}[\![!(t' \multimap t'')]\!] = {} & \{(\{\}, \mathbf{Many}\, v) \mid (\{\}, v) \in \mathcal{V}[\![t' \multimap t'']\!]\} \\ & \cup \{(\{\}, \mathbf{fix}(g, x : t, e : t')) \mid \forall (\sigma', v') \in \mathcal{V}[\![t']\!].\ e[x/v][g/\mathbf{fun}\, x : t \to e_1] \in \mathcal{C}[\![t']\!]\} \\ & \text{where} e_1 = e[g/\mathbf{let}\,\mathbf{Many}\, g = \mathbf{fix}\,(g, x : t, e : t')\,\mathbf{in}\,\mathbf{fun}\, x : t \to e]\end{aligned}$$

$$\mathcal{V}[\![!t']\!] = \{(\{\}, \mathbf{Many}\, v) \mid t \notin \{\mathbf{bool}, \mathbf{int}, \mathbf{elt}, t' \multimap t''\} \land (\{\}, v) \in \mathcal{V}[\![t']\!]\}$$

$$\mathcal{V}[\![\forall fc.\ t]\!] = \{(\sigma, \forall fc.\ v) \mid \forall f.\ (\sigma, v[fc/f]) \in \mathcal{V}[\![t[fc/f]]\!]\}$$

$$\mathcal{V}[\![t' \otimes t'']\!] = \{(\sigma, \langle v', v'' \rangle) \mid \exists \sigma', \sigma''.\ (\sigma', v') \in \mathcal{V}[\![t]\!] \land (\sigma'', v'') \in \mathcal{V}[\![t'']\!] \land \sigma = \sigma' \star \sigma''\}$$

$$\mathcal{V}[\![t' \multimap t'']\!] = \{(\sigma, \mathbf{fun}\, x : t' \to e'') \mid \forall (\sigma', v') \in \mathcal{V}[\![t']\!].\ \sigma = \sigma' \star \sigma'' \text{defined} \Rightarrow (\sigma' \star \sigma'', e''[x/v']) \in \mathcal{C}[\![t'']\!]\}$$

$$\mathcal{C}[\![t]\!] = \{(\sigma_s, e) \mid \forall \sigma_r.\ \sigma_s \star \sigma_r \text{defined} \Rightarrow \exists \sigma_f, v, .(\sigma_s \star \sigma_r, e) \to^n (\sigma_f \star \sigma_r, v) \in \mathcal{V}[\![t]\!]\}$$

$$\mathcal{S}[\![\Delta; \cdot]\!]\theta = \{(\{\}, \delta) \mid \mathrm{dom}(\Delta) = (\delta) \land \forall x \in \mathrm{dom}(\Delta).\ (\{\}, \delta(x)) \in \mathcal{V}[\![\theta(t)]\!]\}$$

$$\mathcal{S}[\![\Delta; \Gamma, x : t]\!]\theta = \{(\sigma \uplus \sigma_x, \delta[x \mapsto v_x]) \mid (\sigma, \delta) \in \mathcal{S}[\![\Delta; \Gamma]\!] \land (\sigma_x, v_x) \in \mathcal{V}[\![\theta(t)]\!]\}$$

$$[\![\Theta; \Delta; \Gamma \vdash e : t]\!] = \forall \theta, \delta, \sigma.\ \mathrm{dom}(\Theta) = \mathrm{dom}(\theta) \land (\sigma, \delta) \in \mathcal{S}[\![\Delta; \Gamma]\!]\theta \Rightarrow \sigma(\delta(e)) \in \mathcal{C}[\![\theta(t)]\!]$$